IN THE CLAIMS

Please amend the claims as follows:

1. (Currently amended) A method for supporting input/output for a virtual machine, comprising,:

executing virtual machine application instructions, wherein the application instructions are executed using micro architecture code of a processor architecture, wherein the micro architecture code includes an instruction interpreter to execute the virtual machine application instructions;

receiving an I/O access from the virtual machine application;

~~using memory protection to generate an exception caused by the I/O access;~~

~~entering a single step mode to perform the I/O access using a host operating system;~~

upon receiving the I/O access, generating an exception;

performing the I/O access by using a host operating system;

updating state data for the virtual machine application in accordance with the I/O access; and

resuming execution of the virtual machine application.

2. (Currently amended) The method of claim 1, wherein the ~~micro architecture code includes an~~ instruction interpreter is further configured to function with an instruction translator to translate target instructions into host VLIW instructions to ~~to execute~~execute the virtual machine application instructions.

3. (Original) The method of claim 1, wherein the micro architecture code includes an instruction translator to execute the virtual machine application instructions.

4. (Original) The method of claim 1, further comprising:

executing a monitor to implement the I/O access from the virtual machine application, wherein the monitor is configured to handle the exception caused by the I/O access.

5. (Original) The method of claim 4, further comprising:

entering a single step mode, wherein the monitor single steps through the application instructions to handle the exception.

6. (Original) The method of claim 4, further comprising:

using the monitor to maintain at least one virtual device to implement the I/O access from the virtual machine application.

7. (Original) The method of claim 6, further comprising:

using the host operating system to access a real device in response to an access to the virtual device; and

updating the state data for the virtual machine application in accordance with I/O data retrieved from the real device.

8. (Original) The method of claim 1, wherein the virtual machine application instructions comprise target instructions and the micro architecture code comprises host instructions.

9. (Original) The method of claim 8, wherein the target instructions are x86 instructions and the host instructions are VLIW instructions.

10. (Original) The method of claim 8, wherein the virtual machine is an x86 compatible virtual machine.

11. (Currently amended) A system for supporting input/output for a virtual machine, comprising,:

a processor architecture including micro architecture code configured to execute natively on a CPU hardware unit of the processor architecture; and

a memory coupled to the processor architecture, the memory storing virtual machine application instructions, wherein the application instructions are executed using the micro architecture code, wherein the micro architecture code includes an instruction interpreter to execute the virtual machine application instructions, the micro architecture code causing the processor architecture to implement a method comprising:

receiving an I/O access from the virtual machine application;

upon receiving the I/O access, generating an exception;

performing the I/O access by using a host operating system;

updating state data for the virtual machine application in accordance with the I/O access; and

resuming execution of the virtual machine application.

12. (Currently amended) The system of claim 11, wherein the instruction interpreter is further configured to function with an instruction translator to translate target instructions into host VLIW instructions ~~micro architecture code includes an instruction interpreter~~ to execute the virtual machine application instructions.

13. (Original) The system of claim 11, wherein the micro architecture code includes an instruction translator to execute the virtual machine application instructions.

14. (Original) The system of claim 1, further comprising:

executing a monitor to implement the I/O access from the virtual machine application, wherein the monitor is configured to handle the exception caused by the I/O access.

15. (Original) The system of claim 14, further comprising:

entering a single step mode, wherein the monitor single steps through the application instructions to handle the exception.

16. (Original) The system of claim 14, further comprising:

using the monitor to maintain at least one virtual device to implement the I/O access from the virtual machine application.

17. (Original) The system of claim 16, further comprising:

using the host operating system to access a real device in response to an access to the virtual device; and

updating the state data for the virtual machine application in accordance with I/O data retrieved from the real device.

18. (Original) The system of claim 11, wherein the virtual machine application instructions comprise target instructions and the micro architecture code comprises host instructions.

19. (Original) The system of claim 18, wherein the target instructions are x86 instructions and the host instructions are VLIW instructions.

20. (Original) The system of claim 18, wherein the virtual machine is an x86 compatible virtual machine.

21. (Currently amended) A computer readable media for implementing support for an input/output process for a virtual machine, the media storing computer readable code which when executed by a processor causes the processor to implement a method comprising:

executing virtual machine application instructions, wherein the application instructions are executed using micro architecture code of a processor architecture, wherein the micro architecture code includes an instruction interpreter to execute the virtual machine application instructions;

receiving an I/O access from the virtual machine application;

upon receiving the I/O access, generating an exception;

performing the I/O access by using a host operating system;

updating state data for the virtual machine application in accordance with the I/O access; and

resuming execution of the virtual machine application.

22. (Currently amended) The computer readable media of claim 21, wherein the ~~micro architecture code includes an~~ instruction interpreter is further configured to function with an instruction translator to translate target instructions into host VLIW instructions to execute the virtual machine application instructions.

23. (Original) The computer readable media of claim 21, wherein the micro architecture code includes an instruction translator to execute the virtual machine application instructions.

24. (Original) The computer readable media of claim 21, further comprising:

executing a monitor to implement the I/O access from the virtual machine application, wherein the monitor is configured to handle the exception caused by the I/O access.

25. (Original) The computer readable media of claim 24, further comprising:

entering a single step mode, wherein the monitor single steps through the application instructions to handle the exception.

26. (Original) The computer readable media of claim 24, further comprising:

using the monitor to maintain at least one virtual device to implement the I/O access from the virtual machine application.

27. (Original) The computer readable media of claim 26, further comprising:

using the host operating system to access a real device in response to an access to the virtual device; and

updating the state data for the virtual machine application in accordance with I/O data retrieved from the real device.

28. (Original) The computer readable media of claim 21, wherein the virtual machine application instructions comprise target instructions and the micro architecture code comprises host instructions.

29. (Original) The computer readable media of claim 28, wherein the target instructions are x86 instructions and the host instructions are VLIW instructions.

30. (Original) The computer readable media of claim 28, wherein the virtual machine is an x86 compatible virtual machine.

31. (cancelled)

32. (cancelled)